UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/675,265	09/29/2003	Kevin Brown	SVL92003045US1	4880

47069 7590 10/31/2007
KONRAD RAYNES & VICTOR, LLP
ATTN: IBM54
315 SOUTH BEVERLY DRIVE, SUITE 210
BEVERLY HILLS, CA 90212

EXAMINER	
TIMBLIN, ROBERT M	
ART UNIT	**PAPER NUMBER**
2167	

MAIL DATE	DELIVERY MODE
10/31/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/675,265
Filing Date: September 29, 2003
Appellant(s): BROWN ET AL.

Janaki K. Davda
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 7/09/2007 appealing from the Office action mailed 2/08/2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,801,905	Andrei	10-2004

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Objections

The Claim 1 objection to the minor informality has been withdrawn. The amendment to claim 1 has been entered.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1-39 are rejected under 35 U.S.C. 102(e) as being anticipated by Andrei (U.S. Patent 6,801,905 B2). In the following passages, figures and reference numbers, Andre teaches:

With respect to claim 1, a method for processing predicates in an iterator function, comprising:

under control of an iterator function processor that executes as part of a data store engine (figure 3 and drawing reference 360), when an iterator function included in a statement is invoked (figure 8A col. 11 line 4-13 and col. 24 line 25-67),

obtaining one or more predicates included in the statement (figures 5 and 8A, at least reference numbers 801 and 802, col. 11 line 3-10, and col. 13, example 1);

applying the one or more predicates to a row of data (col. 17 line 15-25);

if applying the one or more predicates results in a match, returning the row of data (col. 11 line 33-40); and

if applying the one or more predicates does not result in a match, searching for another row of data for which application of the one or more predicates results in a match (col. 11, line 33-40. Therein Andrei teaches retrieving relevant rows).

With respect to claim 2, the method of claim 1, wherein obtaining the one or more predicates comprises:

obtaining a qualification descriptor that describes the one or more predicates and one or more functions that are used to process the one or more predicates (figure 5).

With respect to claim 3, the method of claim 2, wherein each function is used to process one of the predicates (col. 31, line 1-11).

With respect to claim 4, the method of claim 1, wherein each of the one or more predicates comprises a simple predicate (example 1 in column 13).

With respect to claim 5, the method of claim 1, wherein the iterator function is invoked by the data store engine (360) and further comprising: returning the row of data to the data store engine (col. 11, line 33-40).

With respect to claim 6, a method for processing predicates, comprising:

receiving a statement including an iterator function and one or more predicates (figure 8A);

creating a qualification descriptor that describes the one or more predicates and one or more functions that are to be used to evaluate the one or more predicates (figure 5); and

invoking the iterator function one or more times (figure 8B, reference 809), until receiving a done indicator from the iterator function (DONE în figure 8B), wherein each invocation of the iterator function results in receiving either a row of data for which at least one predicate has been applied (col. 11 line 33-40) or the done indicator (DONE in figure 8B) and wherein one or more of the predicates included in the statement that have not been applied by the iterator function are applied by the data store engine (figure 5; applying predicates).

With respect to claim 7, the method of claim 6, wherein the qualification descriptor provides a handle to each of the one or more functions (figure 5 and predicates to apply row).

With respect to claim 8, the method of claim 6, further comprising: when the iterator function is invoked, receiving an indication from the iterator function indicating whether the one or more predicates were applied by the iterator function (fig. 5, applied predicates).

With respect to claim 9, the method of claim 6, further comprising:

when the iterator function is invoked, receiving a row of data from the iterator function that matches the qualification of the one or more predicates (col. 11, line 33-40. Therein Andrei teaches retrieving relevant rows).

With respect to claim 10, the method of claim 9, further comprising:

applying one or more additional predicates to the received row of data, wherein the one or more additional predicates refer to a column of data that is not in a result set generated by the iterator function (figure 5 and col. 6, line 23-41).

With respect to claim 11, the method of claim 9, further comprising:

applying one or more additional predicates to the received row of data, wherein the one or more additional predicates performs a join between two tables (col. 5, line 9-18 and figure 4).

With respect to claim 12, a method for processing predicates, comprising:

under control of a data store engine (360),

receiving a statement including an iterator function and one or more predicates (figure 8A);

creating a qualification descriptor that describes the one or more predicates and one or more functions that are to be used to evaluate the one or more predicates (figure 5); and

invoking the iterator function (figure 8B, reference 809);

under control of an iterator function processor (figure 3, 340),

retrieving the qualification descriptor (figure. 5);

obtaining a row of data that matches at least one of the one or more predicates in the qualification descriptor (col. 3,line 20-27 and col. 11 line 33-40);

setting indicators to indicate which of the one or more predicates have been applied by the iterator function processor (figure. 5, applied predicates); and

returning the row of data to the data store engine (col. 11 line 33-40); and

under control of the data store engine (360),

applying any of the one or more of the predicates included in the statement that have not been applied by the iterator function processor (figure 5; applying predicates).

With respect to claim 13, the method of claim 12, wherein the qualification descriptor describes one or more simple predicates to be applied by the iterator function processor (figure 5).

With respect to claim 14, an article of manufacture including a program for processing predicates in an iterator function, wherein the program causes operations to be performed, the operations comprising:

under control of an iterator function processor that executes as part of a data store engine (360), when an iterator function included in a statement is invoked (figure 3 and drawing reference 360),

obtaining one or more predicates included in the statement (figures 5 and 8A, at least reference numbers 801 and 802, col. 11 line 3-10, and col. 13, example 1);

applying the one or more predicates to a row of data (col. 17 line 15-25);

if applying the one or more predicates results in a match, returning the row of data (col. 11 line 33-40); and

if applying the one or more predicates does not result in a match, searching for another row of data for which application of the one or more predicates results in a match (col. 11, line 33-40. Therein Andrei teaches retrieving relevant rows).

With respect to claim 15, the article of manufacture of claim 14, wherein operations for obtaining the one or more predicates further comprise: obtaining a qualification descriptor that describes the one or more predicates and one or more functions that are used to process the one or more predicates (figure 5).

With respect to claim 16, the article of manufacture of claim 15, wherein each function is used to process one of the predicates (col. 31, line 1-11).

With respect to claim 17, the article of manufacture of claim 14, wherein each of the one or more predicates comprises a simple predicate (example 1 in column 13).

With respect to claim 18, the article of manufacture of claim 14, wherein the iterator function is invoked by a data store engine (360) and wherein the operations further comprise: returning the row of data to the data store engine (col. 11, line 33-40).

With respect to claim 19, an article of manufacture including a program for processing predicates, wherein the program causes operations to be performed, the operations comprising:

receiving a statement including an iterator function and one or more predicates (figure 8A);

creating a qualification descriptor that describes the one or more predicates and one or more functions that are to be used to evaluate the one or more predicates (figure 5); and

invoking the iterator function one or more times (figure 8B, reference 809), until receiving a done indicator from the iterator function (DONE in figure 8B), wherein each invocation of the iterator function results in receiving either a row of data for which at least one predicate has been applied (col. 11 line 33-40) or the done indicator (DONE in figure 8B) and wherein one or more of the predicates included in the statement that have not been applied by the iterator function are applied by the data store engine (figure 5; applying predicates).

With respect to claim 20, the article of manufacture of claim 19, wherein the qualification descriptor provides a handle to each of the one or more functions (figure 5 and predicates to apply row).

With respect to claim 21, the article of manufacture of claim 19, wherein the operations further comprise: when the iterator function is invoked, receiving an

indication from the iterator function indicating whether the one or more predicates were applied by the iterator function (figure 5, applied predicates).

With respect to claim 22, the article of manufacture of claim 19, wherein the operations further comprise: when the iterator function is invoked, receiving a row of data from the iterator function that matches the qualification of the one or more predicates (col. 11, line 33-40. Therein Andrei teaches retrieving relevant rows).

With respect to claim 23, the article of manufacture of claim 22, wherein the operations further comprise: applying one or more additional predicates to the received row of data, wherein the one or more additional predicates refer to a column of data that is not in a result set generated by the iterator function (figure 5 and col. 6, line 23-41).

With respect to claim 24, the article of manufacture of claim 22, wherein the operations further comprise: applying one or more additional predicates to the received row of data, wherein the one or more additional predicates performs a join between two tables (col. 5, line 9-18 and figure 4).

With respect to claim 25, an article of manufacture including a program for processing predicates, wherein the program causes operations to be performed, the operations comprising:

under control of a data store engine (360),

receiving a statement including an iterator function and one or more predicates (figure 8A);

creating a qualification descriptor that describes the one or more predicates and one or more functions that are to be used to evaluate the one or more predicates (figure 5); and

invoking the iterator function (figure 8B, reference 809); and

under control of an iterator function processor (figure 3, 340),

retrieving the qualification descriptor (figure. 5);

obtaining a row of data that matches at least one of the one or more predicates in the qualification descriptor (col. 3,line 20-27 and col. 11 line 33-40); and

setting indicators to indicate which of the one or more predicates have been applied by the iterator function processor (figure. 5, applied predicates);

returning the row of data to the data store engine (col. 11 line 33-40); and

under control of the data store engine (360),

applying any of the one or more of the predicates included in the statement that have not been applied by the iterator function processor (figure 5; applying predicates).

With respect to claim 26, the article of manufacture of claim 25, wherein the qualification descriptor describes one or more simple predicates to be applied by the iterator function processor (figure 5).

With respect to claim 27, a computer system having at least one program for processing predicates in an iterator function, comprising:

under control of an iterator function processor that executes as part of a data store engine(360), when an iterator function included in a statement is invoked,

obtaining one or more predicates included in the statement (figures 5 and 8A, at least reference numbers 801 and 802, col. 11 line 3-10, and col. 13, example 1);

applying the one or more predicates to a row of data (col. 17 line 15-25);

if applying the one or more predicates results in a match, returning the row of data (col. 11 line 33-40); and

if applying the one or more predicates does not result in a match, searching for another row of data for which application of the one or more predicates results in a match (col. 11, line 33-40). Therein Andrei teaches retrieving relevant rows).

With respect to claim 28, the computer system of claim 27, wherein obtaining the one or more predicates comprises: obtaining a qualification descriptor that describes the one or more predicates and one or more functions that are used to process the one or more predicates (figure 5).

With respect to claim 29, the computer system of claim 28, wherein each function is used to process one of the predicates (col. 31, line 1-11).

With respect to claim 30, the computer system of claim 27, wherein each of the one or more predicates comprises a simple predicate (example 1 in column 13).

With respect to claim 31 the computer system of claim 27, wherein the iterator function is invoked by a data store engine (360) and further comprising: returning the row of data to the data store engine (col. 11, line 33-40).

With respect to claim 32, a computer system having at least one program for processing predicates, comprising:

under control of a data store engine (360),

receiving a statement including an iterator function and one or more predicates (figure 8A);

creating a qualification descriptor that describes the one or more predicates and one or more functions that are to be used to evaluate the one or more predicates (figure 5); and

invoking the iterator function one or more times (figure 8B, reference 809), until receiving a done indicator from the iterator function (DONE in figure 8B) wherein each invocation of the iterator function results in receiving either a row of data for which at least one predicate has been applied (col. 11 line 33-40) or the done indicator (DONE in figure 8B) and wherein one or more of the predicates included in the statement that have not been applied by the iterator function are applied by the data store engine (figure 5; applying predicates).

With respect to claim 33, The computer system of claim 32, wherein the qualification descriptor provides a handle to each of the one or more functions (figure 5 and predicates to apply row).

With respect to claim 34. The computer system of claim 32, further comprising: when the iterator function is invoked, receiving an indication from the iterator function indicating whether the one or more predicates were applied by the iterator function (fig. 5, applied predicates).

With respect to claim 35, the computer system of claim 32, further comprising: when the iterator function is invoked, receiving a row of data from the iterator function that matches the qualification of the one or more predicates (col. 11, line 33-40. Therein Andrei teaches retrieving relevant rows).

With respect to claim 36 the computer system of claim 35, further comprising: applying one or more additional predicates to the received row of data, wherein the one or more additional predicates refer to a column of data that is not in a result set generated by the iterator function (figure 5 and col. 6, line 23-41).

With respect to claim 37, the computer system of claim 35, further comprising: applying one or more additional predicates to the received row of data, wherein the one or more additional predicates performs a join between two tables (col. 5, line 9-18 and figure 4).

With respect to claim 38, a computer system for processing predicates, comprising:

under control of a data store engine (360),

means for receiving a statement including an iterator function and one or more predicates (figure 8A);

means for creating a qualification descriptor that describes the one or more predicates and one or more functions that are to be used to evaluate the one or more predicates (figure 5); and

means for invoking the iterator function (figure 8B, reference 809); and

under control of an iterator function processor (figure 3, 340),

means for retrieving the qualification descriptor (figure. 5);

means for obtaining a row of data that matches at least one of the one or more predicates in the qualification descriptor (col. 3,line 20-27 and col. 11 line 33-40);

means for setting indicators to indicate which of the one or more predicates have been applied by the iterator function processor (figure. 5, applied predicates);

and means for returning the row of data to the data store engine (col. 11 line 33-40); and

under control of the data store engine (360),

means for applying any of the one or more of the predicates included in the statement that have not been applied by the iterator function processor (figure 5; applying predicates).

With respect to claim 39, the computer system of claim 38, wherein the qualification descriptor describes one or more simple predicates to be applied by the iterator function processor (figure 5).

(10) Response to Argument

Appellant's arguments filed in the Brief have been fully considered but they are not persuasive.

Appellant argues on pages 7-11 of the Brief that the Andrei reference does not anticipate the claimed aspect of "when an iterator function included in a statement is invoked" as presented in claim 1. The Examiner disagrees for the reasons provided below:

As indicated in the Office Action of 2/8/07, Andrei teaches receiving SQL statements from the client (col. 11 line 4-5). A statement sent to a database for accessing the database is common practice in the art. Figure 8A correspondingly shows receiving a statement (step 801). Further, and in the next step, the received statement is parsed and converted into a logical operator tree (step 802) that represents the components of the query (statement) (col. 11 line 9-10). The processing is accomplished by engine (360) (col. 11 line 5, Andrei).

The Examiner submits that in the Appellant's use of the phrase "when an iterator function included in a statement is invoked" one could find similar meaning of "invoke" to equate to other interpretations such as when an iterator is called upon, brought forth, brought into existence, etc. Andrei explicitly states bringing about and recognizing an iterator in a statement (and thus the iterator is invoked). For example, Andrei discloses creating an "iterator" tree (col. 6 line 23-27) for a query execution plan. Further in this

passage, Andrei discloses the query execution plan is created when a statement is received and that this plan covers the total semantics of the query (col. 6 line 36).

As another example of "invoking" an iterator, Andrei explicitly discloses optimization time descriptions of [Volcano] runtime iterators (col. 24 line 30-32). Here it is seen that the Volcano iterators would *have* to be invoked (i.e. brought forth from a statement) in order to provide a description of them.

The Appellant further argues that Andrei does not show that when an iterator function included in a statement is invoked, obtains one or more predicates included in the statement (page 10, 3rd full paragraph of the Brief). In response, the Examiner submits that Andrei gives a clear example of the processing of a statement including an iterator function (that is invoked by being "brought forth") and predicates. Specifically, example 1 (col. 13) shows a statement that is received (i.e. the "select" indicates the beginning of an SQL statement[1]). The next line after select shows a function sum(13) which is equivalent to an iterator function. In the 'where' clause is predicates (i.e. qualifications that correspond to applicants definition of qualification; see page 8 line 8 of Brief) p1-p5 (e.g. c2='automotive' - p3, o3<date 'yesterday'-p4). The Examiner submits that the statement in example 1 is processed to yield a descriptive table of logical properties in figure 5. In figure 5, it is seen that the predicates included in the statement are obtained. Specifically, and for example, Equivalence class {I} includes iterator function (i.e. aggregation functions row) sum(13). Predicates from the

[1] In database practice, an SQL statement follows the syntax of SELECT [data] FROM [table] WHERE [insert predicates to qualify/return rows here].

statement (example 1, col. 13) are therefore obtained and described (i.e. applied or yet to be applied are descriptors of the predicates/qualification) in the table of figure 5.

Similarly, and in response to the argument pertaining to claim 2 (page 11 of brief), figure 5 shows the predicates that are obtained (i.e. predicates p1-p5) and the function that included the predicates (i.e. the aggregate functions).

Similarly, and in response to the argument pertaining to claim 3 (page 12, first full paragraph of the Brief), Andrei's figure 5 discloses the aggregate functions that are used to process the one or more predicates (i.e. p1-p5).

Similarly, and in response to the argument pertaining to claim 4 (page 12, 2nd full paragraph), Andrei discloses the iterator function under control of an engine (360, figure 3) in a database server (i.e. suggesting a data store engine). Furthermore, Andrei describes retrieving relevant information (e.g. row 355) (col. 11, line 37-39) to suggest returning a row of data to the data store engine. The Examiner submits that Andrei's retrieving relevant information (i.e. retrieving a relevant row) is the same thing as the claimed returning a row of information.

The Appellant further agues (top of page 14 of the Brief) regarding claim 6 that Andrei does not describe that at least one predicate has been applied by the iterator function and that one or more of the predicates included in the statement that have not been applied by the iterator function are applied by the data store engine. The Examiner disagrees because, again in figure 5, Andrei discloses the predicates (p1-p5) that have been obtained have been either applied (i.e. the "applied predicates" row) or predicates yet to apply (i.e. the "predicates to apply" row). As these are predicates of

the iterator function (i.e. the aggregation functions in figure 5), Andrei clearly shows the predicates applied by the iterator function. For example, predicate p5 has been applied in the equivalence class {I}. Predicates p1-p4 are to be applied in the equivalence class {I}. Further, the Examiner submits that one can easily construe that the predicates to apply are *going* to be applied (i.e. the predicates to apply are applied and then listed as applied) and thus sufficiently teach the one or more of the predicates included in the statement that have not been applied by the iterator function are applied."

Further figure 5 describes the claimed done indicator on the last row where the completion of an aggregation function (i.e. iterator) is complete (yes) or not complete (no). The examiner submits that teaching whether the function is complete or not corresponds to Applicant's done indicator and thus the aggregation function is executed (i.e. invoked) until finished (i.e. done).

In response to Appellant's arguments concerning claim 10 (page 14, 3rd paragraph), the Examiner submits that Andrei discloses predicates that refer to a column of data that is not in a result set generated by the iterator function. Again, figure 5 illustrates the logical properties that describe a given statement (i.e. example 1, col. 13). Further described, the logical properties describe the characteristics of an expression (col. 17 line 1-3). Also, in column 17 lines 13-15, Andrei discloses applying predicates on projected columns. The notion of Andrei having predicates still to be applied on projected columns suggests that the predicates to be applied refer to a column not in the result set (i.e. a projected column is one that has not been processed yet and therefore not in the result set).

Appellant further argues (page 14, 2nd to the last paragraph) regarding claim 11 that Andrei does not disclose one or more additional predicates performs a join between two tables. The Examiner disagrees because column 5, line 15 describes a sub-plan joining tables A and B. The Examiner submits that a joining of two tables, commonly performed in by a join predicate (that is included in a statement), sufficiently teaches the additional predicate that performs a join on two tables. Furthermore, figure 4 shows the equivalence classes (i.e. Eqc{c, o, l} of figure 5) that specify the joining of two tables. For example, the predicates p1, p2 in the equi-joins row (figure 5) show joining predicates for equivalence class {c}.

Lastly, Appellant argues (end of page 15 of Brief) in regard claim 12, 25, and 38 that Andrei fails to show receiving a statement including an iterator function and one or more predicates. As seen in the above arguments, Andrei discloses this limitation at least in the example 1 of column 13. That is a statement with an iterator function and one or more predicates is received and described in the table of figure 5. The Examiner further submits, in light of the above rejection and arguments that the Andrei reference describes the interaction between a data store engine an iterator function processor in processing an iterator function.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Robert M. Timblin

Patent Examiner AU 2167

Primary Examiner Art Unit 2167 for

Conferees: John Cottingham Eddie Lee

EDDIE C. LEE
SUPERVISORY PATENT EXAMINER